

Санкт-Петербургский государственный университет
Математическое обеспечение и администрирование информационных
систем

Технология программирования

Хайдаршин Александр Марсельевич

Разработка распределенной информационной системы для
управления сведениями о результатах интеллектуальной
деятельности: клиентская часть, сбор статистики, создание
отчетов

Бакалаврская работа

Научный руководитель:
профессор кафедры информатики, д. ф.-м. н., доц. Тулупьев А. Л.

Рецензент:
с. н. с. лаб. ТиМПИ СПИИРАН, к. ф.-м. н. Суворова А. В.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY
Software and Administration of Information Systems

Technology of Programming

Aleksandr Khaydarshin

Development of distributed information
system for managing information of
intellectual activity results: client side,
statistics collection, report creation

Bachelor's Thesis

Scientific supervisor:
Dc. Sc. in Math, Assoc. Prof., Prof. Computer Science Department Alexander Tulupyeu

Reviewer:
Ph. D. in Math, Senior Researcher Alena Suvorova

Saint-Petersburg
2017

Оглавление

Введение	5
1. Обзор предметной области	8
1.1. Mendeley	8
1.2. Scopus	10
1.3. Google Scholar	11
1.4. ResearchGate	12
1.5. eLIBRARY.RU	13
1.6. Вывод	13
2. Описание инструментария	15
2.1. Программное обеспечение и веб-сервисы	15
2.1.1. Bitbucket	15
2.1.2. SourceTree	15
2.1.3. WebStorm	16
2.1.4. Koala	16
2.2. Фреймворки	17
2.2.1. AngularJS	17
2.2.2. Materialize	18
2.3. Используемые библиотеки	18
2.3.1. Lodash	18
2.3.2. PapaParse	19
2.3.3. VScroll	19
3. Проектирование системы	20
3.1. Функциональность и требования	20
3.2. Защита данных	21
4. Программная реализация	22
4.1. Структура клиентской части	22
4.1.1. Навигационная панель	22
4.1.2. Уведомления о результатах действий	23

4.2.	Модальные окна аутентификации пользователя	24
4.3.	Страница профиля	25
4.3.1.	Редактирование информации о профиле	25
4.4.	Страница управления публикациями	26
4.4.1.	Боковая панель	27
4.4.2.	Модальное окно добавления публикации	28
4.4.3.	Модальное окно редактирования публикации	28
4.4.4.	Модальное окно удаления публикаций	29
4.4.5.	Экспорт выбранных публикаций	29
4.4.6.	Таблица научных публикаций	29
4.5.	Импорт публикаций	30
4.5.1.	Выбор языка публикации	30
4.5.2.	Сопоставление категории публикации с образцом	31
4.6.	Страница администратора	32
4.6.1.	Контент на странице администратора	32
4.6.2.	Возможности администратора	33
4.7.	Работа в группах	34
4.7.1.	Создание группы	34
4.7.2.	Вступление в группу	35
5.	Пример работы с веб-приложением	36
	Заключение	39
	Приложение	41
	Список литературы	49

Введение

Научным работникам, исследователям и иным творческим деятелям необходимо управлять сведениями о своих результатах интеллектуальной деятельности в сфере исследований и разработок. В случае малого количества публикаций, их учёт возможно проводить вручную, но при большом количестве достижений структурировать информацию становится сложно.

Рассмотрим несколько реальных примеров, когда необходимо предоставить список научных трудов, более подробно:

- пользователю необходимо написать автореферат диссертации на соискание ученой степени кандидата наук. В этом случае, на последней странице (или ближе к концу) авторы приводят списки своих публикаций по теме диссертации;
- лаборатории необходимо составить годовой отчет, в котором должны быть списки достижений, в том числе список публикаций от группы исследователей, работающих в лаборатории;
- научным сотрудникам может понадобиться список опубликованных статей, монографий и иных интеллектуальных работ при заполнении бланка расчета индивидуального рейтингового показателя (ИРП) результативной научной деятельности, чтобы получить надбавку к заработной плате;
- при подаче сведений для автоматизированной системы учета результатов интеллектуальной деятельности Российской академии наук (АСУ РИД) лаборатории требуется предоставить публикации сотрудников;
- для выступления в качестве официального оппонента и подачи отзыва по кандидатской диссертации также необходимо предоставить список опубликованных научных работ;

- необходимы ссылки и в собственных публикациях, в том числе и на свои работы. В данном случае пристатейный список оформляется в виде списка литературы.

В теории, количество работ в сфере исследований, необходимых к предъявлению, может быть неограничено. Очевидно, что с такими списками вручную работать затруднительно. Необходимо также учитывать и то, что для разных ситуаций требуются списки в разных форматах (например, .doc или .xls), а также разных типов (например, в виде списка литературы или таблицы).

Нельзя не сказать, что не существует похожих систем, предназначенных для работы с публикациями. Но существующие системы имеют ряд недостатков и не покрывают все возникающие потребности. Поэтому создается система, позволяющая автоматизировать работу со списком публикаций.

Разрабатываемая информационная система RADOMS (Research And Development Outcomes Management System) [28], предназначенная для автоматизации редактирования списка публикаций и создания выборки в различных форматах по задаваемым фильтрам, состоит из базы данных для хранения информации о пользователях и их публикациях, а также из веб-приложения, позволяющего работать с данными из базы.

Работа с публикациями подразумевает внесение данных о публикации в базу с помощью ручного заполнения форм или путем импорта из заранее подготовленного EXCEL файла в формате CSV/XLS/XLSX, редактирование данных, а также экспортирование их из базы в соответствии с запросом пользователя в различных форматах на выбор, например, в DOC/DOCX, EXCEL(CSV/XLS/XLSX).

Целью проекта RADOMS является автоматизация управления сведениями о результатах интеллектуальной деятельности. Целью настоящей работы является реализация клиентской части приложения в рамках проекта по созданию вышеописанной системы. В частности, поставлены следующие задачи: создание страниц веб-приложения, форм и модальных окон, импорт списка публикаций из файла, сбор статисти-

ки о сервисе, хранение данных в пределах сессии, реализация групповой работы.

1. Обзор предметной области

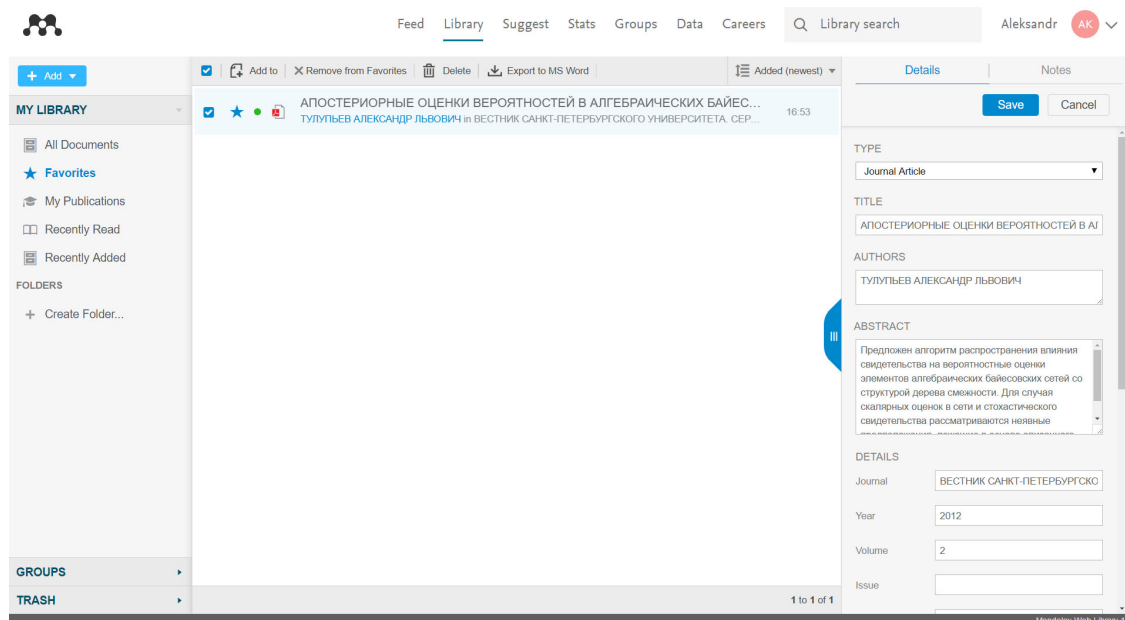
В данном разделе рассматриваются и сравниваются аналоги разрабатываемой системы.

К возможным конкурентным сервисам можно отнести такие системы, как Mendeley [14], Scopus [20], Google Scholar [7], ResearchGate [18], eLIBRARY.ru [27], которые, однако, имеют ряд недостатков и отсутствующих функций. В связи с этим актуальной задачей является разработка более гибкой системы, которая соответствовала бы всем поставленным требованиям.

Рассмотрим отличия указанных систем более подробно.

1.1. Mendeley

Условно-бесплатная программа, а также веб-сервис, позволяющий хранить и просматривать исследовательские труды в формате PDF; присутствуют элементы социальной сети. Платный доступ позволяет расширить доступный объем облачного хранилища для загрузки публикаций.



К основным функциям системы можно отнести следующие:

- хранение работ в формате PDF, из которых автоматически извлекаются метаданные о публикации;

- ручное добавление сведений о публикации;
- синхронизация с сервисом Scopus;
- возможность работать с текстами документов;
- поиск по общей базе публикаций;
- коллективная работа с документами в общей библиотеке;
- экспорт перечня публикаций.

Ключевые отличия от разрабатываемой системы:

- регистрация возможна только с помощью электронной почты;
- веб-версия системы имеет ограниченный функционал. Например, экспорт списка публикаций может быть только в формате .xml, когда в десктопной версии доступны также форматы .bib и .res;
- автоматическое извлечение метаданных из файлов публикаций, написанных на русском языке, не работает, поэтому все данные необходимо вносить вручную;
- формат вывода нельзя настраивать, отсутствует возможность экспорта по фильтру в веб-версии;
- доступен импорт списка публикаций с помощью файлов формата .bib, .ris, .xml;
- часть возможностей, например, хранение самих файлов является избыточным, поскольку сервис RADOMS необходим для удобной работы со списком публикаций.

RADOMS является веб-сервисом, поэтому весь функционал системы доступен с любого устройства, а авторизация возможна и с помощью социальных сетей, что ускоряет вход в систему. Сохранить результат работы можно в форматах .doc/docx и .csv/xls/xlsx, настроив перед этим фильтр.

1.2. Scopus

Scopus — библиографическая и реферативная база данных и инструмент для отслеживания цитируемости статей, опубликованных в научных изданиях: журналах, книгах, сборниках статей с конференций. Данный сервис предназначен только для работы (и достаточно ограниченной) с публикациями в изданиях, входящих в Scopus. Таким образом, исследователь не сможет работать со всеми своими публикациями, если часть из них не содержится в базе Scopus. Основной функционал системы является платным.

Tulup'ev, Alexander L.
St. Petersburg Institute for Informatics and Automation, Russian Academy of Sciences, Theoretical and Interdisciplinary Computer Science Lab (TICS Lab), Saint Petersburg (ex Leningrad), Russian Federation
Author ID: 13608565400

Documents: 15
Citations: 8 total citations by 8 documents
h-index: 2
Co-authors: 26
Subject area: Computer Science, Mathematics View More

15 Documents | Cited by 8 documents | 26 co-authors

15 documents View in search results format

Export all to RIS file Add all to list Set document alert Set document feed

Choose your default reference manager or file type: 2

Scopus offers integrated export functionality with Mendeley and Refworks. Or, to use a different reference manager, choose a file format.

Save to Mendeley RIS Format EndNote, Reference Manager CSV Excel BibTeX Text ASCII in HTML

Choose the information to export: Choose the information you want to export to the reference manager or file.

Citation information only Citation information only Citations and abstract information Citations, abstract and references All available information Specify fields to be exported

Export

Author	Year	Title	Citations
Tulup'ev, A.V., Tulup'ev, A.L.	2016	Proceedings of the 19th International Conference on Soft Computing and Measurements, SCM 2016	0
Tulup'ev, A.V., Levenets, D.G., Tulup'ev, A.L.	2016	Proceedings of the 19th International Conference on Soft Computing and Measurements, SCM 2016	0
Tulup'ev, E.A., Berezin, A.I., Tulup'ev, A.L.	2016	Advances in Intelligent Systems and Computing	0
Tulup'ev, A., Abramov, M., Tulup'ev, T.	2016	Advances in Intelligent Systems and Computing	0
Azarov, A., Abramov, M., Tulup'ev, A., Tulup'eva, T.	2016	Advances in Intelligent Systems and Computing	0


Основные отличия от сервиса RADOMS:

- доступ к базе данных является платным;
- публикации добавляются автоматически, редактирование недоступно;
- экспорт публикаций возможен в форматах .ris, .csv, .bib и в текстовом формате. Вид выходного файла можно настраивать только с помощью фильтров, в то время, как RADOMS предлагает экс-

порт результатов в виде списка литературы или табличного списка публикаций.

1.3. Google Scholar

Google Scholar — бесплатная поисковая система по текстам публикаций различных форматов. Выполняется поиск не только по статьям, доступным в открытом доступе, но и по публикациям, доступным только в библиотеках или платных сервисах. Результатом поиска являются ссылки на статьи, но в силу ограничений платных сервисов не любую публикацию можно просмотреть полностью.



Leonov GA
[St.Petersburg State University](#)
[Mathematics, Mechanics](#)
 Verified email at math.spbu.ru - [Homepage](#)

[Follow](#)


Title	1–20	Cited by	Year
Устойчивость нелинейных систем с неединственным состоянием равновесия <small>АХ Гелиг, ГА Леонов, ВА Якубович</small>	570 *	1978	
Frequency-domain methods for nonlinear analysis: theory and applications <small>GA Leonov, DV Ponomarenko, VB Smirnova World Scientific</small>	338	1996	
Hidden attractors in dynamical systems. From hidden oscillations in Hilbert–Kolmogorov, Aizerman, and Kalman problems to hidden chaotic attractor in Chua circuits <small>GA Leonov, NV Kuznetsov International Journal of Bifurcation and Chaos 23 (01), 1330002</small>	308	2013	
Stability of stationary sets in control systems with discontinuous nonlinearities <small>VA Yakubovich, AK Gelig, GA Leonov World Scientific</small>	297	2004	
Localization of hidden Chua's attractors <small>GA Leonov, NV Kuznetsov, VI Vagaitsev</small>	285	2011	

Google Scholar

[Get my own profile](#)

Citation indices

	All	Since 2012
Citations	8382	5183
h-index	38	34
i10-index	170	109



Co-authors [View all...](#)

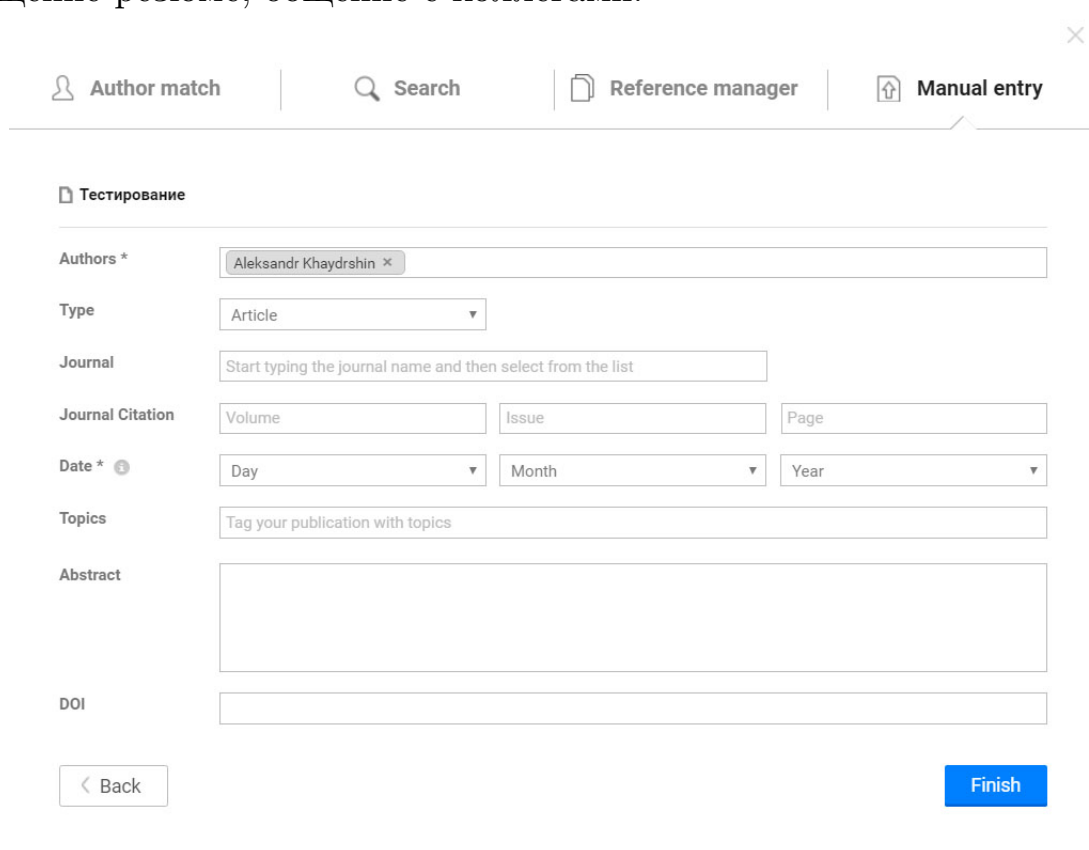
- Nikolay V. Kuznetsov
- Seledzhi SM
- MV Yuldashev
- Renat Yuldashev
- Pekka Neittaanmäki
- Alexander Pogromsky
- Alexander Fradkov (Александр Фрадк...
- Kuznetsova OA
- Awadhesh Prasad
- Knnstantin Starkov

Отличия:

- невозможно импортировать список публикаций из файла, но добавление вручную присутствует;
- нет возможности групповой работы со списком публикаций;
- настройка фильтров для экспорта результатов отсутствует;
- экспорт возможен в форматах .enw, .ris, .csv, .bib, структуру файла изменить нельзя.

1.4. ResearchGate

ResearchGate — бесплатная социальная сеть для учёных различных научных дисциплин. Имеется собственный поисковый движок по научным трудам из общей базы публикаций, благодаря которому профиль пользователя может формироваться автоматически, без участия конечного пользователя. Среди возможностей системы можно выделить обмен полными текстами публикаций, сбор статистики о поисковых запросах других пользователей, затрагивающих конечного автора, размещение резюме, общение с коллегами.



The screenshot shows the 'Manual entry' tab of the ResearchGate interface. At the top, there are navigation links: 'Author match', 'Search', 'Reference manager', and 'Manual entry'. Below these, the form is titled 'Тестирование'. It contains several input fields: 'Authors *' with a dropdown menu showing 'Aleksandr Khaydrshin'; 'Type' with a dropdown menu showing 'Article'; 'Journal' with a text input field and a hint 'Start typing the journal name and then select from the list'; 'Journal Citation' with three input fields for 'Volume', 'Issue', and 'Page'; 'Date *' with three dropdown menus for 'Day', 'Month', and 'Year'; 'Topics' with a text input field and a hint 'Tag your publication with topics'; 'Abstract' with a large text area; and 'DOI' with a text input field. At the bottom left is a '< Back' button, and at the bottom right is a blue 'Finish' button.

Отличия от разрабатываемого сервиса:

- ResearchGate является в первую очередь социальной сетью, что затрудняет основную задачу сервиса RADOMS — работу со списком публикаций;
- несмотря на то, что формирование профиля пользователя возможно и без его участия, для просмотра данной информации необходима регистрация;

- нельзя корректировать автоматически сформированную информацию о публикациях;
- нельзя сформировать список публикаций.

1.5. eLIBRARY.RU

Еще одним инструментом для работы с публикациями является eLibrary.ru — электронная библиотека научных публикаций, интегрированная с Российским индексом научного цитирования (РИНЦ) [29]. На портале доступны статьи из журналов с открытым доступом, но также имеется и платный доступ по подписке для организаций. Присутствует возможность создания и корректировки списка научных публикаций и цитирований в РИНЦ, Web of Science [24] и Scopus.

Отличия от сервиса RADOMS:

- отсутствует возможность загружать список своих работ из заранее подготовленного документа, но есть ручной ввод данных для авторизованных представителей организаций;
- регистрация через социальные сети невозможна;
- отсутствие адаптивной верстки, что делает работу с сервисом на мобильных устройствах некомфортной;
- возможность настроить вид результирующего файла о результатах интеллектуальной деятельности отсутствует.

1.6. Вывод

Несмотря на то, что часть функций, которые хотелось видеть в сервисе, реализованы в других проектах, единой системы, отвечающей всем требованиям не существует. В следствие данного факта было решено разрабатывать собственный проект. Учтем особенности и преимущества существующих сервисов: групповую работу с публикациями, экспорт в различных форматах и видах, доступность на любых

устройствах. Вместе с тем, будет предложен новый функционал, который будет решать ряд поставленных задач:

- авторизация через социальные сети;
- адаптивный дизайн;
- импорт списка научных трудов из файла;
- удобная работа со списком публикаций;
- экспорт файлов в соответствии со стандартом ГОСТ.

2. Описание инструментария

Программное обеспечение было выбрано таким образом, чтобы отвечать современным тенденциям веб-разработки. Опишем весь выбранный инструментарий подробно.

2.1. Программное обеспечение и веб-сервисы

2.1.1. Bitbucket

Поскольку разработка данного сервиса является командным проектом, одной из задач являлся выбор веб-сервиса, основанного на системе контроля версий, для размещения репозитория. В качестве соответствующего решения был выбран сервис Bitbucket [2], поддерживающий систему контроля версий Git [5].

Данный сервис аналогичен по функциям GitHub [6], но, в частности, предоставляет возможность создания бесплатного “закрытого”¹ репозитория, в то время, как у GitHub данная возможность является платной. Bitbucket поддерживает систему bug-трекинга, что позволяет сообщать другим разработчикам об ошибках в разрабатываемой системе и указывать степень критичности данной ошибки; также возможно создание задач куратором проекта и присвоение данной задачи конкретному разработчику.

2.1.2. SourceTree

Для удобной работы с системой контроля версий Git была выбрана бесплатная графическая оболочка SourceTree [21] от компании Atlassian, которой также принадлежит Bitbucket, что обеспечивает полную интеграцию всех возможностей. Данная оболочка упрощает взаимодействие с репозиторием, но при этом является мощным инструментом. Позволяет создавать и копировать репозитории, осуществлять коммиты, обновления и объединения веток, разрешать конфликты, работать с историей

¹“Закрытым” репозиторием в данном случае будем считать репозиторий, доступ к которому есть только у определенного круга людей, в данном случае — у разработчиков и координатора проекта

изменений.

2.1.3. WebStorm

В качестве среды разработки был выбран продукт от компании JetBrains — WebStorm [25]. Данный инструмент имеет совместимость с технологиями, выбранными для реализации разработки настоящего сервиса: AngularJS [1], Node.js [16]. IDE (интегрированная среда разработки, от англ. Integrated Development Environment) обеспечивает автодополнение, анализ кода “на лету”, удобную навигацию по исходному коду программы, отладку и другие функции.

Плюсом данного продукта является его бесплатность для студентов.

2.1.4. Koala

Работа с файлами .CSS [3] в больших проектах является неудобной: нет возможности использовать переменные, отсутствуют вложенные правила, куски кода зачастую приходится дублировать, выходной файл разрастается и становится нечитаемым. SASS [19] позволяет элегантно решить данную проблему для разработчиков, но компиляцию в обычный .CSS файл, который “понимают” браузеры необходимо проводить с помощью программных средств.

В сравнении с другим препроцессорным языком LESS для CSS, SASS имеет некоторые преимущества. Например, присутствуют логические и циклические операторы; также различается способ написания переменных: в LESS используется символ @, который уже используется в CSS для медиа-запросов и блоков @keyframes, а символ \$, используемый SASS, не имеет в CSS своего значения.

Для работы с SASS-файлами подходит бесплатная кроссплатформенная программа Koala, позволяющая компилировать соответствующие файлы в обыкновенный CSS.

Koala отслеживает изменения файлов в реальном времени и автоматически компилирует их в фоновом режиме, отслеживает ошибки, имеет простой минималистичный интерфейс. При этом, есть встроен-

ная возможность компрессии выходного кода, а также поддержка различных компиляторов.

2.2. Фреймворки

В качестве набора программного обеспечения, используемого для разработки сервиса был выбран MEAN (аббревиатура от MongoDB [15], Express.js [4], AngularJS, Node.js) стек. Все компоненты данного стека поддерживают программирование на JavaScript [10], что облегчает разработку. Данный набор инструментов является популярным и актуальным решением для веб-приложений, подобных по характеру разрабатываемому сервису.

2.2.1. AngularJS

Фреймворком для клиентской части был выбран AngularJS — JavaScript фреймворк с открытым исходным кодом, предназначенный для создания одностраничных веб-приложений². Использует MVC-шаблон³.

Особенностью данного фреймворка является двустороннее связывание, позволяющее динамически изменять представление при изменении данных модели AngularJS, и наоборот. Поддерживаются шаблоны, пользовательские директивы, маршрутизация и иные функции, необходимые для разработки современных веб-приложений.

Рассмотрим элементы данного фреймворка, которые используются при разработке сервиса:

- модули — основная форма хранения различных частей приложения: контроллеров, сервисов, фильтров, директив и т.д. Модули ассоциируют конкретный участок html-страницы с приложением AngularJS;

²Одностраничным веб-приложением будем считать приложение, использующее единственный HTML [8] документ как оболочку для остальных страниц, которые динамически загружаются.

³MVC-шаблон (от англ. Model, View, Controller) — схема разделения данных приложения, интерфейса пользователя, а также управляющей логики на три отдельных компоненты.

- контроллеры — предназначены для связывания модели данных и html-кода;
- сервисы — объекты или функции, доступные из любого места приложения, реализующие общие задачи;
- директивы — являются расширением html синтаксиса. Предназначены, например, для валидации форм, обработки событий, генерации шаблонов;
- \$scope объект — выполняет роль модели приложения. В нем определяются любые объекты, которые можно использовать в итоговом представлении html-кода и к которым возможна привязка данных.

На момент начала работы над данным сервисом AngularJS имел стабильную версию 1.5.x, но в сентябре 2016 года вышла стабильная версия Angular 2.0, которая не имеет обратной совместимости. Важным фактом является то, что функциональности версии, используемой в проекте (v1.5.5) хватает для выполнения поставленных задач, поэтому переписывать исходный код — неэффективно.

2.2.2. Materialize

В качестве CSS фреймворка был выбран Materialize [13], в котором описаны принципы “материального дизайна” от Google [12]. Предоставляет большой набор CSS и JavaScript компонентов для использования, позволяет создавать адаптивный дизайн для любого устройства.

2.3. Используемые библиотеки

2.3.1. Lodash

Lodash [11] — библиотека JavaScript, реализующая дополнительные функции для работы с массивами, объектами, функциями, позволяющие упростить код и имеющие аналоги в функциональных языках программирования. С помощью этой библиотеки становятся доступными к

использованию такие функции, как `map`, `clone`, `filter`, `findIndex`, `orderBy` и другие.

2.3.2. PapaParse

Данная библиотека является JavaScript CSV парсером, позволяет конвертировать данные из CSV в JSON-объект [9] и наоборот. Из основных преимуществ перед конкурентами стоит выделить многопоточность, что позволяет парсить файлы большого объема без сбоев браузера. PapaParse [17] обрабатывает данные на стороне пользователя без необходимости передачи данных через интернет.

2.3.3. VScroll

VScroll [22] для AngularJS позволяет отображать и обрабатывать лишь те DOM [23] элементы, которые видны пользователю в текущий момент, вместо обработки всего списка данных, что положительно сказывается на производительности при работе с очень большими коллекциями данных.

3. Проектирование системы

3.1. Функциональность и требования

В соответствии с техническим заданием, были поставлены следующие задачи:

- создание соответствующих страниц веб-приложения, форм и модальных окон:
 - главная страница;
 - модальное окно аутентификации;
 - страница профиля;
 - модальное окно редактирования профиля;
 - страница публикаций;
 - модальные окна для добавления, редактирования, экспорта публикаций;
 - страница администратора.
- импорт публикаций в базу данных:
 - добавление публикаций путем ручного ввода данных или с помощью подготовленного файла в формате .csv, .xls или .xlsx;
 - при добавлении или импорте публикации необходимо учитывать, что она может быть на английском языке. В этом случае необходимо корректно выводить информацию при экспорте данных;
 - при импорте публикаций из пользовательского файла возможна ситуация, когда категория публикации пользователя отличается от заданных в системе (например, статья в журнале, монография, статья на конференции). В таком случае необходимо сообщить пользователю о данном событии, а также запрашивать выбор категории из предложенного списка.

- страница администратора:
 - на странице администратора необходимо обеспечить корректную подгрузку данных о пользователях в случае, если выбрано табличное отображение;
 - реализация возможности сбрасывать пароли выбранных пользователей, а также удалять их.
- необходимо осуществить возможность создания групп и получение их списка в соответствии с API (от англ. Application Programming Interface), реализованным на серверной части;
- реализация хранения данных в пределах сессии пользователя;
- обновление библиотек, связанных с компилятором экспортируемых данных;
- различные незначительные исправления.

3.2. Защита данных

Для защиты паролей пользователей было использовано шифрование. Реализация данного функционала осуществлена использованием библиотеки `bcrypt` из менеджера пакетов `npm` для Node.js [26]. Данная библиотека позволяет генерировать хэш⁴ пароля на стороне сервера внутри пользовательской модели.

При генерации хэша пароля также используется так называемая «соль» — строка случайных данных, которая подается на вход хеш-функции вместе с исходными данными, что позволяет повысить сложность восстановления исходного пароля, из-за наличия новых случайно сгенерированных данных в пароле пользователя.

⁴Хэширование — преобразование входных данных произвольной длины в выходную битовую строку фиксированной длины, а обратное преобразование при этом невозможно.

4. Программная реализация

4.1. Структура клиентской части

Разработанный сервис является одностраничным веб-приложением. На всех функциональных страницах приложения доступна навигационная панель вверху страницы и информация о сервисе в footer (“подвале”) страницы, а остальная часть является динамически изменяемой в зависимости от действий пользователя.

В связи с этим, в главный файл веб-приложения `index.html` помимо мета-информации внесены все скрипты для работы сервиса, ссылки на страницы сервиса, соответствующие им CSS файлы, меню, а также контроллеры и модули AngularJS, доступные на всех страницах: контроллер навигации, уведомлений, расположения элементов. Также с помощью директивы шаблонов `ng-include` реализовано добавление модальных окон для аутентификации пользователя. Ключевым является наличие атрибута `ng-app`, что делает модуль `app` корневым для всего приложения. (См. Листинг 1 в приложении)

С помощью метода `config` модуля `app` определены маршруты внутри сервиса. Для их конфигурации использован объект `$routeProvider`, принимающий два параметра: название и объект маршрута. Объект маршрута состоит из представления, а также контроллера, отвечающего за него. В случае, если у пользователя нет прав для доступа к определенной странице, или набран неверный `url`-адрес, производится перенаправление на страницу с информацией о сервисе. Таким образом, страницам сервиса назначены соответствующие им контроллеры.

4.1.1. Навигационная панель

Навигационная панель для разных пользователей принимает разный вид: неавторизованному пользователю доступна кнопка для входа в систему, авторизованные пользователи имеют возможность управлять сведениями о результатах научной деятельности и редактировать свой профиль, а администраторам сервиса доступна также вкладка со

статистикой. Для обеспечения данного функционала создан сервис `user`, доступный из любой части приложения. В данном сервисе описана модель пользователя на клиентской части, а также реализованы функции, обращающиеся к API на серверной части сервиса, проверяющие роль пользователя.

Рассмотрим на примере функцию, которая с сервера запрашивает информацию о том, является ли пользователь администратором (функция проверки авторизации пользователя аналогична данной за исключением url-адреса соответствующего API). В данной функции используется механизм **Deferred**-объектов. Данные объекты предоставляют методы управления состоянием выполнения асинхронной операции, как рассматриваемый в примере запрос. При успешном выполнении вызывается метод `deferred.resolve`, при ошибке `deferred.reject`, а `deferred.promise` возвращается в качестве результата. (См. Листинг 2 в приложении)

За связывание модели данных и html-кода навигационной панели отвечает контроллер `authorizationController`. Данный контроллер передает объекту `$scope.user` данные, определенные в вышеописанном сервисе `user`, а также имеет функции для открытия модального окна аутентификации и выхода из системы. С помощью директив `ng-if` реализовано скрытие или отображение соответствующих роли пользователя пунктов меню.

4.1.2. Уведомления о результатах действий

Для отображения информационных сообщений о результатах действия пользователя реализован сервис `errorModel`. В данном сервисе описаны массивы объектов, отвечающих за информационные сообщения, сообщения об ошибках и сообщения о корректности действий пользователя; функция, распределяющая события по данным массивам и очищающая их, и функция удаления конкретного события из массива событий. Также реализован сервис для логирования ошибок.

В контроллере, отвечающем за уведомления, представлена модель, состоящая из вышеописанных массивов, и методов, описанных в сервисе `errorModel`. С помощью директивы `ng-repeat` реализован обход мо-

дели событий для отображения уведомлений определенной категории, а также действия с уведомлениями: закрытие одного уведомления или сразу всех, если их несколько. Как и в случае с пунктами меню, отображение уведомлений реализовано с помощью директивы `ng-if` при условии, что модель событий не пуста.

4.2. Модальные окна аутентификации пользователя

Как уже было сказано, если пользователь не прошел аутентификацию, ему доступна страница с описанием информации о сервисе, а также возможность вызвать модальное окно входа в учетную запись. Также реализованы модальные окна для регистрации в системе, восстановления пароля, входа с помощью социальных сетей при условии, что они привязаны к учетной записи на странице профиля.

Для всех модальных окон аутентификации создан единый контроллер `modalsController`. В данном контроллере определена модель пользователя, состоящая из полей, доступных при регистрации: `email`, пароль, имя, фамилия и т.д., а также присутствует логическое поле `rememberMe`, необходимое для сохранения сессии; описаны функции инициализации соответствующих модальных окон. Для валидации `email` при регистрации реализована функция, выполняющая запрос к базе данных на наличие данного `email` в ней. При регистрации необходимо дважды ввести пароль — совпадение введенных данных также проверяется в данном контроллере.

Рассмотрим функцию, которая отправляет запрос о регистрации на сервер. В объекте `userInfo` описаны данные, которые передаются соответствующему API. В случае положительного ответа на данный запрос сообщение об этом отображается пользователю с помощью метода `push` сервиса `errorModel`, а модальное окно закрывается. (См. Листинг 3 в приложении)

В контроллере описаны функции для сброса пароля, авторизации через пару `email`-пароль и социальные сети, и для повторной отправки верификационного письма пользователю, поскольку после регистрации

необходимо подтвердить учетную запись.

Между данными модальными окнами реализована перекрестная навигация, а подтвердить действия нельзя, пока все поля форм не будут корректно заполнены. После успешной аутентификации происходит перенаправление на страницу профиля.

4.3. Страница профиля

Для просмотра информации об учетной записи, а также для изменения личных данных создана страница профиля и модальное окно для его редактирования.

В контроллере, отвечающем за страницу профиля, определена модель пользователя, функция, запрашивающая сведения из базы данных, и функция для открытия модального окна редактирования профиля, которое добавлено на данную страницу с помощью директивы шаблонов `ng-include`.

Пользователю доступна следующая информация о своей учетной записи: имя и фамилия, место работы/учебы, должность, количество публикаций в системе и количество созданных отчетов. Также возможен просмотр расширенной информации: страны, номера телефона и даты регистрации; присутствует возможность привязать социальные сети для быстрого входа в систему.

4.3.1. Редактирование информации о профиле

На данной странице доступна форма для редактирования внесенных при регистрации сведений, а также возможность смены пароля путем указания нового в двух полях. Контроллер, отвечающий за связь между моделью данных и представлением окна редактирования профиля, содержит функцию `update`. Внутри данной функции создается PUT-запрос к серверу для обновления данных, а в случае, если изменен и пароль, то вложенный POST-запрос отправляет соответствующие данные.

4.4. Страница управления публикациями

Для управления сведениями о результатах интеллектуальной деятельности создана соответствующая страница и модальные окна добавления, редактирования, удаления соответствующих сведений и экспорта выборки в форматах docx и xls/xlsx/csv. Страница состоит из двух ключевых элементов: меню слева от публикаций с возможными действиями и фильтрацией результирующего списка, а также таблицы со списком научных работ (колонки в таблице можно скрывать или показывать с помощью выпадающего списка в правом нижнем углу страницы).

The screenshot displays the 'RADOMS: R&D Outcomes Management System' interface. At the top, there is a navigation bar with links: 'О ПРОЕКТЕ', 'УПРАВЛЕНИЕ ПУБЛИКАЦИЯМИ' (active), 'ПРОФИЛЬ', and 'ВЫХОД'. Below the navigation bar, the left sidebar contains a search section with 'Публикации: выбрано 0 из 10', 'Поиск в таблице', and 'Поиск в атрибутах'. The sidebar also features a list of filters: 'Категория' (with sub-filters 'Доклад на конференции' and 'Статья в журнале'), 'Авторы', 'Год', 'Метка', 'Журнал', 'Номер/выпуск', and 'Том'. The main content area shows a table of publications with columns: 'Категория', 'Авторы', 'Название', and 'Год'. The table lists 10 publications, including reports and journal articles, with their respective authors and titles. A dropdown menu in the bottom right corner of the table allows for column visibility toggling.

Категория	Авторы	Название	Год
Доклад на конференции	Хайдаршин А.М., Зотов М.А., Суворова А.В., Тулупьев А.Л.	РАСПРЕДЕЛЕННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА RADOMS. КЛИЕНТСКАЯ ЧАСТЬ, СБОР СТАТИСТИКИ, СОЗДАНИЕ ОТЧЕТОВ	2017
Статья в журнале	Зотов М.А., Левенец Д.Г., Тулупьев А.Л., Золотин А.А.	Синтез вторичной структуры алгебраических байесовских сетей: инкрементальный алгоритм и статистическая оценка его сложности	2016
Доклад на конференции	Levenets D.G., Zotov M.A., Romanov A.V., Tulupyyev A.L., Zolotin A.A., Filchenkov A.A.	Decremental and Incremental Reshaping of Algebraic Bayesian Networks Global Structures	2016
Доклад на конференции	Berezin A.I., Ivanova A.V., Zotov M.A.	Minimal Join Graphs' Set Synthesis: Performance Statistical Estimate of the Decremental Algorithm	2016
Доклад на конференции	Berezin A.I., Romanov A.V., Zotov M.A.	Algebraic Bayesian Networks Secondary Structure Synthesis: All possible Minimal Join Graphs' Set Capacity Estimate	2016
Доклад на конференции	Зотов М.А., Тулупьев А.Л.	Вторичная структура алгебраических байесовских сетей: статистическая оценка сложности прямого алгоритма синтеза	2015
Статья в журнале	Зотов М.А., Тулупьев А.Л., Сироткин А.В.	Статистические оценки сложности прямого и жадного алгоритмов синтеза вторичной структуры алгебраических байесовских сетей	2015
Доклад на конференции	Зотов М.А., Тулупьев А.Л.	Синтез вторичной структуры алгебраических байесовских сетей: сравнительный анализ статистических оценок сложности двух алгоритмов	2015
Статья в журнале	Левенец Д.Г., Зотов М.А., Тулупьев А.Л.	Инкрементальный алгоритм синтеза минимального графа смежности	2015
Статья в журнале	Зотов М.А., Тулупьев А.Л.	Синтез вторичной структуры алгебраических байесовских сетей: методика статистической оценки сложности и компаративный анализ прямого и жадного алгоритмов	2015

Рис. 1: Внешний вид страницы управления публикациями

Для работы на данной странице создан контроллер `postController`, являющийся родительский по отношению к остальным. В данном контроллере определены внутренняя модель, объект, отвечающий за представление, реализованы функции инициализации модальных окон, получения списка публикаций, его обновления, сопоставления категории публикации, редактирования данных, сброса фильтров поиска, сброса выделения публикаций. При этом, внутренняя модель использует данные из сервиса `publicationModel`, в котором описаны атрибуты ко-

лонок, категорий и полей для этих категорий, необходимые данные для маппинга и поиска по атрибутам. Данный контроллер также использует сервис `columnChooser` для отображения выбранных колонок, сервис `compiler` для компиляции файла отчета, сервис `publicationHelper` подготавливающий документ для отчета в нужный вид, библиотеку `VScroll` для снижения нагрузки при отображении большого количества публикаций.

4.4.1. Боковая панель

Для быстрого поиска научных работ по фильтрам, составленным из мета-данных о ней, и управления wybranнми работами создана боковая панель, которую при необходимости возможно свернуть.

В верхней части данной панели представлена информация о количестве выбранных публикаций из всего списка, а также поля для фильтрации выборки по возможным значениям атрибутов и таблице. Поиск по атрибутам выполняется при помощи фильтра, а поиск по таблице подразумевает сокрытие неподходящих работ, что реализовано в функции `updatePublicationsList`. Помимо этого, внутри данной функции происходит обновление количества подходящих работ у атрибутов в боковой панели, если фильтрация исключает часть работ из всего списка (См. Листинг 4 в приложении). При выборе какого-либо атрибута из списка также происходит обновление списка работ.

Затем расположены кнопки для открытия модальных окон. Открыть окно добавления публикации возможно всегда, но окна для редактирования, удаления и экспорта публикаций доступны только в том случае, если пользователем выделена хотя бы одна работа. При наличии созданных отчетов ссылки на скачивание документов располагаются в выпадающем меню.

С помощью директивы шаблонов подключена страница с возможными атрибутами для поиска в таблице. Данные атрибуты отображаются, если количество подходящих публикаций не равно нулю, а отображение выполнено обходом по массиву внутренней модели с помощью директивы `ng-repeat`.

4.4.2. Модальное окно добавления публикации

В окне добавления публикации пользователю предлагается внести данные вручную, либо загрузить их из файла. Для удобства добавлена всплывающая подсказка о виде файла, распознаваемым системой.

При импорте сведений о научных работах производится парсинг файла. Элементы первой строки становятся ключами созданного объекта, а данные последующих строк — значениями соответствующих ключей. Пользователю отображены распознанные поля из загруженного документа напротив которых необходимо сопоставить ручную поля из модели, если системе не удалось найти аналог.

В случае, если в загруженном файле недостаточно данных о работе в соответствии с моделью, определенной для данной категории, пользователю предлагается вручную заполнить пропуски. Данная форма аналогична по своему содержанию форме ручного внесения сведений, поэтому далее не будет делаться различий.

Выбор категории публикации реализован с помощью выпадающего меню с заданными в системе значениями. С помощью директивы `ng-repeat` в соответствии с выбранной категорией работы предложены к заполнению поля из сервиса `publicationModel`. Для полей, подразумевающих автодополнение (например, авторы), реализовано отображение возможных значений атрибута в зависимости от введенных данных.

Реализованный контроллер `createPublicationController` содержит функции, связанные с маппингом, определением валидности публикации, автодополнением значений. Функция сохранения внесенных данных реализована следующим образом: на сервер отправляется POST-запрос, содержащий данные о публикациях, а пользователю отображается информационное сообщение.

4.4.3. Модальное окно редактирования публикации

Окно редактирования публикации аналогично окну добавления публикаций вручную. В контроллере реализована функция, отправляющая PUT-запрос о новых данных. По завершении запроса пользователю

отображается информационное сообщение, а модальное окно закрывается.

4.4.4. Модальное окно удаления публикаций

В окне удаления публикаций представлено число выбранных работ к удалению, а также имеются кнопки для подтверждения и отмены операции. Для подтверждения использована функция, реализованная в контроллере, которая отправляет в DELETE-запросе id публикаций, подлежащих удалению, после чего из таблицы удаляются соответствующие записи.

4.4.5. Экспорт выбранных публикаций

В сервисе реализован экспорт выбранных работ в формате docx двух типов: табличного отчета и списка публикаций; а также в формате `xlsx`/`csv` в виде отчета по категориям публикаций или списка публикаций по желаемым колонкам. В последнем случае, а также при указании авторов доступно выпадающее меню с соответствующими возможными значениями данных атрибутов. При выборе экспорта выборки в формате `xlsx` в виде отчета по категориям в результирующий файл вносятся сведения о работах в каждой из категорий, представленных на разных листах.

Создан контроллер, отвечающий за настройку вышеописанных параметров. Компиляция результатов выполняется с помощью функций сервиса `compiler`. Например, для экспорта табличного отчета в формате `docx` в функцию `docxClusterReport` переданы выбранные публикации, выбранные авторы, краткие названия категорий, категории, а также функция, добавляющая отчет в список результатов для загрузки. (См. Листинг 5 в приложении)

4.4.6. Таблица научных публикаций

Основную часть страницы при работе со списком публикаций занимает таблица. Чтобы снизить нагрузку на отрисовку элементов таб-

лицы при значительном их количестве используется сторонний сервис VScroll, поэтому в html-коде страницы к таблице были добавлены соответствующие директивы.

Отображение элементов в таблице реализовано с помощью директивы AngularJS `ng-repeat`, с помощью директивы `ng-bind-html` выполнено связывание данных для заполнения таблицы, директивой `ng-if` выполнено отображение колонок, выбранных пользователем, а с помощью синтаксиса фильтров выполняется выделение текста в колонках таблицы, если поиском выявлено соответствие. Данные для отображения находятся в объекте представления.

Выбор колонок для отображения в таблице предоставлен пользователю с помощью выпадающего меню в правом нижнем углу страницы. Данные о колонках предоставляются сервисом `columnChooser`, а отображаются так же при помощи директив `ng-repeat` и `ng-bind-html`.

4.5. Импорт публикаций

4.5.1. Выбор языка публикации

В модель публикации в базе данных, содержащую сведения об авторе, названии, категории, количестве страниц и т.д., внесены дополнения, позволяющие различать язык, на котором они написаны. По умолчанию считается, что язык публикации — русский. (См. Листинг 6 в приложении)

Пользователю предоставлена возможность изменения поля, отвечающего за язык публикации. Для этого на страницу с импортом или добавлением публикации был добавлен соответствующий чекбокс, отметка которого вносит изменения в базу данных.

Поскольку при добавлении публикации, в зависимости от её категории, пользователю должны быть доступны различающиеся поля или чекбоксы для заполнения, то необходимо корректное их отображение. Данный функционал реализован при помощи директивы `ng-repeat` AngularJS, которая производит обход массива и генерирует соответствующие html-элементы. Соответственно, для каждого типа публика-

ции чекбокс с выбором языка был внесен в список доступных атрибутов. (См. Листинг 7 в приложении)

После того, как пользователь помечает, что публикация на английском языке, экспорт данной публикации производится с соответствующими языковыми изменениями. Например, вместо русского сокращения “С.”, обозначающего “Страница”, отображается сокращение “Р.” от слова “Page”.

Для этого в файл, отвечающий за экспорт, внесены следующие изменения: при добавлении к строке с публикацией какого-либо текста проводится проверка её языка. Например, так будет выглядеть функция, добавляющая количество страниц. (См. Листинг 8 в приложении)

Соответствующие изменения были проведены во всех функциях, отвечающих за результат экспортируемого файла.

4.5.2. Сопоставление категории публикации с образцом

Как уже было сказано в постановке задач, если при импорте публикации из файла веб-приложение автоматически не сможет распознать её категорию, то пользователь не увидит никакой информации о ней, поскольку от категории работы зависят остальные атрибуты, отображаемые пользователю. Данная ситуация является критической, поэтому пользователю сообщается об ошибке, а также предлагается указать категорию вручную.

Для распознавания категории публикации реализована функция `getPublicationCategory`, возвращающая название категории в случае успешного распознавания, а если распознать категорию научной работы не удалось, то функция возвращает ложное значение. (См. Листинг 9 в приложении)

С помощью директивы `ng-if` сообщение об ошибке отображается только в том случае, если применен маппинг данных из пользовательского файла с полями, используемыми в БД для хранения публикаций, а соответствия пользовательской категории с заданными в системе не обнаружено. С помощью выражения `AngularJS` выполнено отображение “проблемной” категории публикации. (См. Листинг 10 в приложении)

Добавление публикации

☒ Загрузить из файла ⓘ

ФАЙЛ публикации.xlsx

✓ Показывать валидные публикации

ДОБАВИТЬ Публикация < 1 > из 2

Не найдено соответствия Вашей категории **Стат в журнале** с категориями, представленными в системе.
Выберите наиболее подходящую для продолжения

Категория публикации

Выберите категорию публикации

- Статья в журнале
- Краткое сообщение в журнал
- Статья в сборнике
- Монография

ДОБАВИТЬ ПУБЛИКАЦИИ ОТМЕНИТЬ ИМПОРТ

Рис. 2: Результат сообщения об ошибке

В результате выполненных действий, пользователю доступно информационное сообщение и возможность выбора категории публикации вручную.

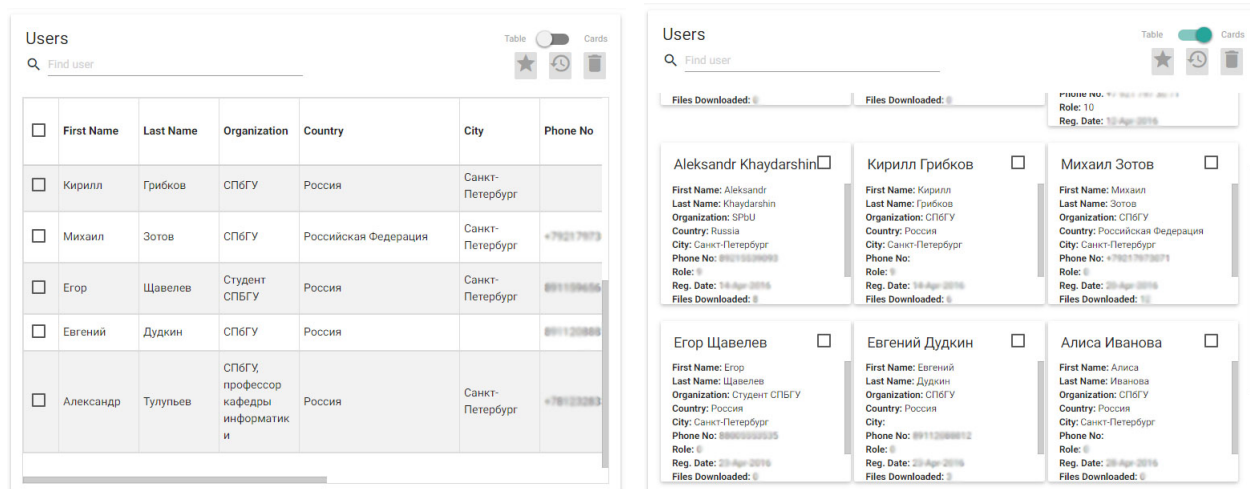
4.6. Страница администратора

Для сбора статистики о разрабатываемом сервисе создана страница администратора. Помимо отображения сведений о пользователях, администратору доступна возможность сбрасывать пароли выбранным пользователям, изменять роль, а также блокировать их.

4.6.1. Контент на странице администратора

Создание страницы администратора принципиально не отличается от других страниц. Но доступ к данной страницы был ограничен от посторонних пользователей, не наделенных соответствующими правами. Для этого на сервере реализована функция `isAdmin`, а отображение соответствующего пункта меню выполнено с помощью атрибута `ng-if`. Отображение информации о пользователях, в соответствии с техниче-

ским заданием, должно быть в виде карточек под каждого зарегистрированного пользователя или в виде таблицы.



Табличное отображение

Отображение карточками

Рис. 3: Разный вид представления данных

В случае табличного отображения, для повышения производительности при работе с большим объемом данных выбрана библиотека VScroll. Для этого в html код страницы добавлена соответствующая директива vscroll к элементу с полосой прокрутки, директива vscroll-port-y (прокрутка по вертикали) к прокручиваемому элементу, связан контекст с директивами и настроен фильтр. (См. Листинг 11 в приложении)

Информация, которая доступна администраторам, состоит из данных, указанных пользователями при регистрации, кроме их паролей; также доступна информация о количестве созданных публикаций и отчетов.

4.6.2. Возможности администратора

Администраторам сервиса доступны возможности по изменению роли пользователей, сбросу паролей выбранных пользователей, а также блокировка их учетных записей. Стоит заметить, что при удалении пользователя, удаляются также и все его публикации. Соответствующие действия выполняются в модальных окнах, создание которых происходит с помощью возможностей AngularJS (См. Листинг 12 в при-

ложении). В остальном, код модальных окон не отличается от кода обычных html-страниц.

Реализация функционала, например, для сброса паролей происходит в соответствии с API, описанным на сервере. Соответствующему API передаются данные о email адресе пользователя, затем `$q.all` объединяет promise-объекты в один, который будет разрешен в случае разрешения всех promise. Для остальных функций исходный код аналогичен данному. (См. Листинг 13 в приложении)

4.7. Работа в группах

Для коллективной работы со списком публикаций, когда пользователю доступны к отображению сведения о научных трудах других пользователей, была придумана идея работы в группах. Данный функционал находится на стадии тестирования и не присутствуют в production-версии продукта.

4.7.1. Создание группы

Пользователи могут самостоятельно создавать группы путем ввода названия, ссылки на группу, а также краткого их описания. Название группы должно быть от 2 до 100 символов, длина ссылки имеет те же ограничения, что и название, но при этом проводится валидация по следующему паттерну: `[[a - zA - Z]]`. Данным паттерном проводится проверка на наличие в ссылке только латинских букв произвольного регистра. Описание группы должно состоять от 50 до 200 символов и, как и в случае с названием, может иметь в себе любые символы. Данный функционал реализован с помощью API, которому передаются вышеописанные данные, а пользователь по умолчанию становится администратором созданной группы. (См. Листинг 14 в приложении)

Форма для создания группы является модальной. Исходный html-код страницы описан в файле `group-create.html`, а инициализация же самого модального окна происходит с помощью фреймворка аналогично другим модальным окнам.

4.7.2. Вступление в группу

Пользователям доступны несколько вариантов для вступления в группу: путем поиска по названию, либо после перехода по специальной ссылке-инвайту, полученной от другого пользователя, состоящего в данной группе. Администраторы групп могут как подтвердить вступление пользователя, так и отказать ему. Данный функционал реализован в соответствии с API, описанным на серверной части приложения.

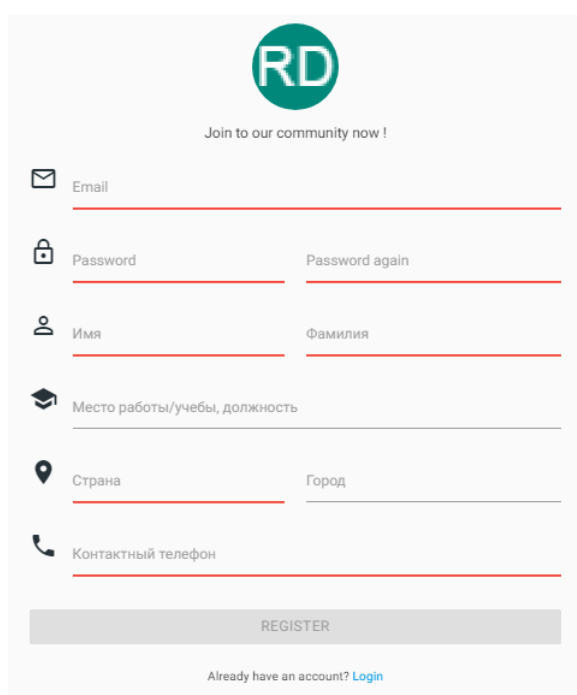
После вступления в группу пользователю доступна информация о количестве участников в ней, а также отображение публикаций всех участников из данной группы на странице управления публикациями. При желании, пользователь может самостоятельно покинуть группы, в которых он состоит.

5. Пример работы с веб-приложением

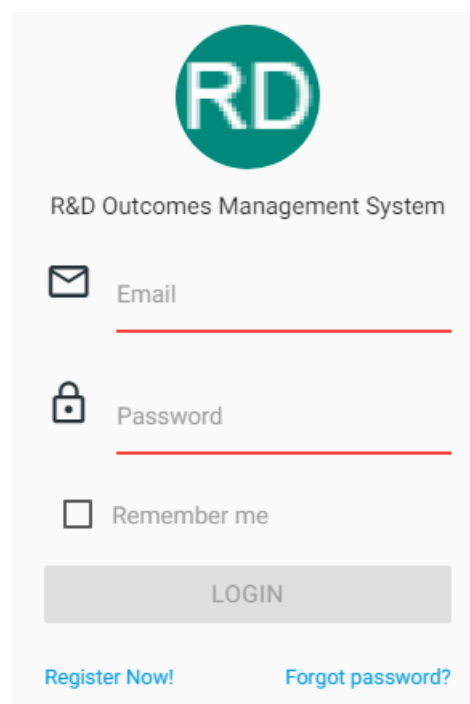
Рассмотрим работу сервиса RADOMS на примере обычного пользователя.

Незарегистрированный пользователь видит информацию о проекте. При нажатии на кнопку “ВХОД” открывается окно аутентификации, куда, для доступа к сервису, можно ввести email и пароль, указанный при регистрации, а также восстановить забытый пароль или зарегистрироваться. Доступна опция “Запомнить меня”.

При регистрации есть обязательные для заполнения поля: email, пароль, имя, фамилия, страна, контактный телефон. Необязательными к заполнению являются такие поля, как место работы и город. После заполнения данных, пользователю необходимо подтвердить регистрацию путем перехода по ссылке, отправленной на указанный email. В случае, если письмо не доставлено, можно запросить повторную отправку.

The registration form features the RAD logo at the top with the text "Join to our community now!". Below this, there are input fields for "Email", "Password", and "Password again". Further down, there are fields for "Имя" (Name) and "Фамилия" (Surname). Below these are fields for "Место работы/учебы, должность" (Work/Study place, position) and "Страна" (Country) and "Город" (City). At the bottom, there is a field for "Контактный телефон" (Contact phone) and a "REGISTER" button. A link "Already have an account? Login" is located at the very bottom.

Форма регистрации

The login form features the RAD logo at the top with the text "R&D Outcomes Management System". Below this, there are input fields for "Email" and "Password". There is a checkbox labeled "Remember me". A "LOGIN" button is positioned below the password field. At the bottom, there are two links: "Register Now!" and "Forgot password?".

Форма авторизации

Рис. 4: Окна аутентификации пользователя

После успешной аутентификации пользователь попадает на страницу управления публикациями. Изначально предлагается добавить публикацию вручную или путем загрузки подготовленного файла.

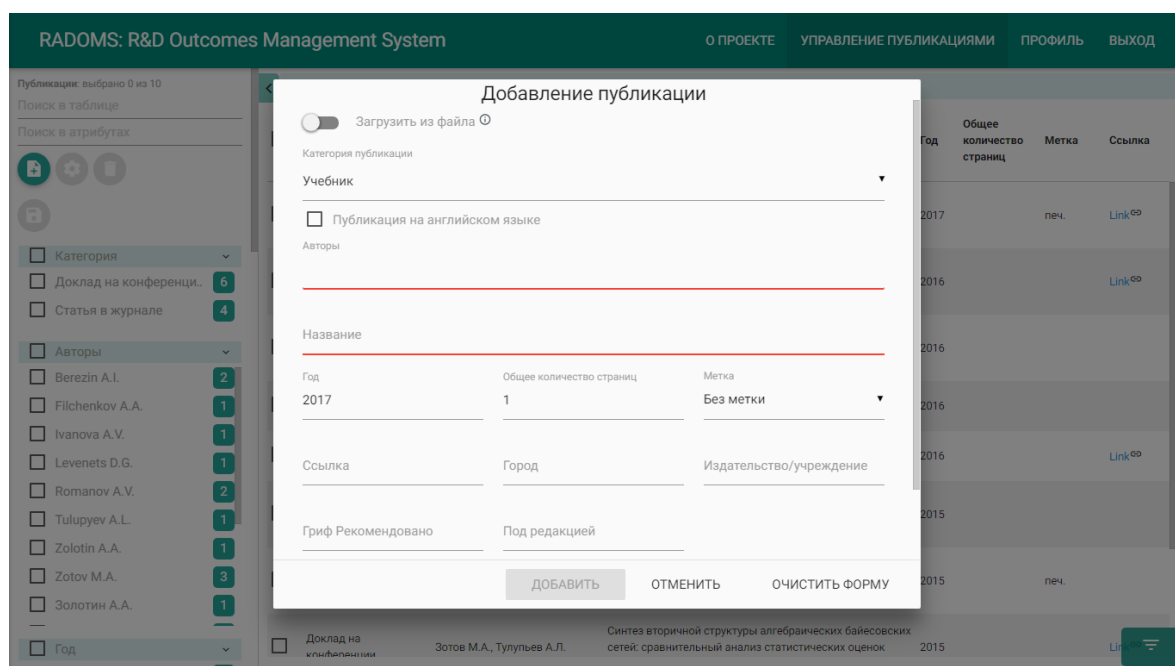


Рис. 5: Пример добавления публикации категории “Учебник”

При наличии публикаций, пользователь может проводить с ними следующие действия: редактировать информацию или удалять их из списка, выполнять поиск в таблице и атрибутах научных трудов, а также управлять внешним видом таблицы публикаций путем выбора соответствующих колонок.

Выбрав одну или несколько публикаций, пользователю становится доступно их сохранение в форматах .doc или .xls на выбор:

- при выборе формата .xls можно выбрать тип отчета: экспорт списка публикаций или отчет по категориям публикаций. В первом случае доступны для редактирования колонки, которые будут в конечном документе;
 - формат выходного документа в обоих случаях: csv или xlsx на выбор.
- при выборе формата .doc доступны следующие типы отчета: сводный отчет (табличный) и список публикаций;
 - в сводном отчете необходимо выбрать авторов, по которым выполняется отчет;

- в списке публикаций можно выбрать стиль написания авторов: обычный, жирный, курсив, курсив + жирный.

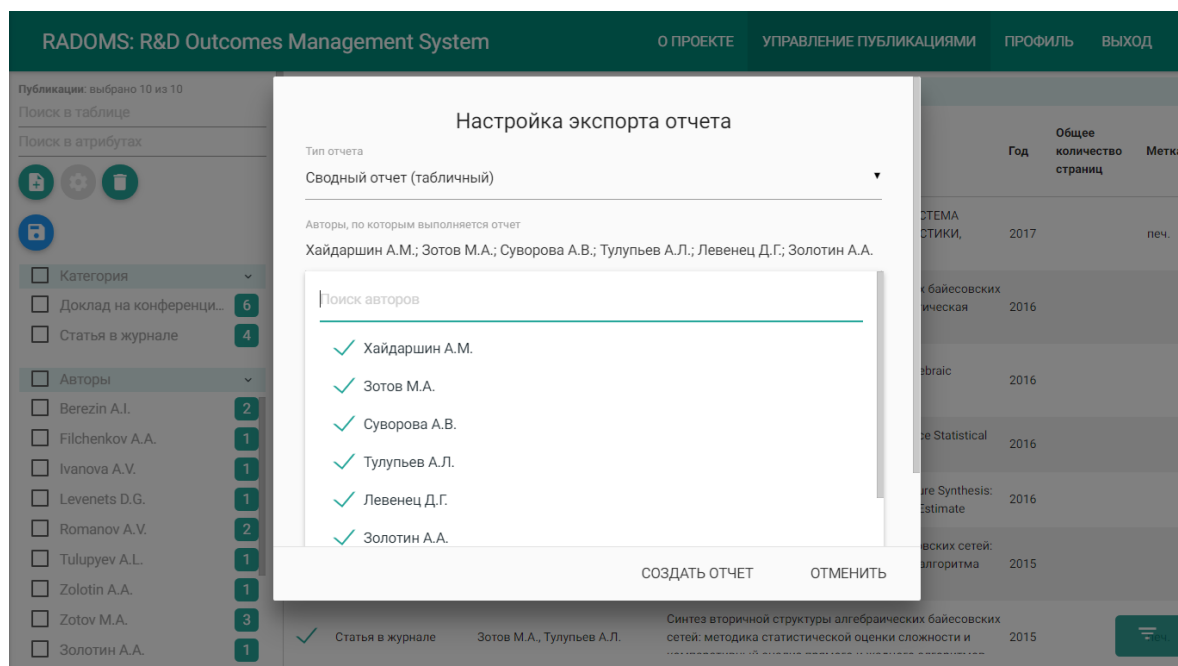


Рис. 6: Пример экспорта выборки в формате .doc в виде сводного отчета

После выбора необходимых опций система подготавливает документ, который пользователь может загрузить на своё устройство.

Помимо работы со сведениями о результатах интеллектуальной деятельности, пользователь может просматривать, а также редактировать данные своего профиля. На соответствующей странице можно изменить данные, указанные при регистрации, включая пароль. Также на данной странице возможно создание групп путем указания названия, ссылки, краткого описания, а также их поиск. Членство в группе позволяет пользователю видеть и экспортировать публикации других участников.

Завершение пользовательской сессии происходит после нажатия на кнопку “ВЫХОД”.

Заключение

В ходе работы над данным проектом:

- опубликована статья на конференции СПИСОК-2016: Грибков К.В., Хайдаршин А.М., Суворова А.В., Тулупьев А.Л. Проект RADOMS: программные компоненты серверной части // Материалы 6-й всероссийской научной конференции по проблемам информатики СПИСОК-2016. (26–29 апреля 2016 г. Санкт-Петербург). СПб.: ВВМ, 2016. С. 463–476;
- принята в печать статья: Хайдаршин А.М., Зотов М.А., Суворова А.В., Тулупьев А.Л. Распределенная информационная система RADOMS. Клиентская часть, сбор статистики, создание отчетов // Материалы 7-й всероссийской научной конференции по проблемам информатики СПИСОК-2017. (26-28 апреля 2017 г. Санкт-Петербург). СПб: ВВМ, 2017;
- направлена заявка на регистрацию системы в РОСПАТЕНТ.

В рамках данной бакалаврской работы были достигнуты следующие результаты:

- рассмотрены существующие аналоги разрабатываемой системы и выявлены недостатки этих аналогов;
- проведен обзор инструментов, необходимых для выполнения поставленных задач;
- сформированы требования к функциональности комплекса;
- созданы веб-страницы и модальные окна для различных действий внутри сервиса;
- создана страница администратора со статистикой о системе;
- реализованы функции для импорта и экспорта публикаций;

- реализованы функции создания и вступления в группу для совместной работы.

Возможные дальнейшие пути развития комплекса:

- создание Telegram бота, компилирующего список публикаций пользователя;
- разнообразие выходных форматов файлов при экспорте.

Приложение

Листинг 1: Структура index.html

```
1 <!DOCTYPE html>
2 <html ng-app="app" class="ng-scope">
3 <head>
4   <title>RADOMS: R&D Outcomes Management System</title>
5   <!-- meta and links "css" -->
6   ...
7 </head>
8 <body>
9 <nav ng-controller="authorizationController">
10   <!-- Menu and href -->
11   ...
12 </nav>
13
14 <div ng-controller="toastsController" class="toast-list"
15   id="toast-container" ng-cloak>
16   <!-- Success messages -->
17   ...
18   <!-- Errors messages -->
19   ...
20   <!-- Info messages -->
21   ...
22 </div>
23
24 <div class="page" ng-controller="layoutController"
25   ng-class="{ 'fixed-body': state.isFixedBody }">
26   <div class="page-ng-view" ng-view=""></div>
27 </div>
28
29 <div ng-include="'components/authorization-modals/login.html'"></div>
30 <div ng-include="'components/authorization-modals/login-social.html'"></div>
31 <div ng-include="'components/authorization-modals/join.html'"></div>
32 <div ng-include="'components/authorization-modals/password-reset.html'"></div>
33
34 <footer class="page-footer">
35   ...
36 </footer>
37
38 <!-- Vendor libraries, Application Files, Directives and Services, Models -->
39 ...
40 </body>
41 </html>
```

Листинг 2: Реализация функции isAdmin на клиентской части

```
1 var isAdmin = function () {
2     var deferred = $.defer();
3     user.inProgress = true;
4
5     $http({
6         method: 'GET',
7         url: '/api/isAdmin'
8     }).then(function (response) {
9         $timeout(function () {
10             user.isAdmin = response.data.isAdmin;
11         });
12
13         if (response.data.isAdmin)
14             deferred.resolve();
15         else {
16             deferred.reject();
17             $location.url('/about');
18         }
19     })['finally'](function (err) {
20         user.inProgress = false;
21         if (err) {
22             user.isAdmin = false;
23             deferred.reject();
24             $location.url('/about');
25         }
26     });
27 });
28
29 return deferred.promise;
30 };
```

Листинг 3: Реализация функции создания аккаунта в сервисе

```
1 $scope.join = function () {
2     $scope.state.join = true;
3     var userInfo = {
4         email: $scope.model.emailNew,
5         password: $scope.model.password,
6         name: $scope.model.name,
7         surname: $scope.model.surname,
8         organization: $scope.model.organization,
9         country: $scope.model.country,
10        city: $scope.model.city,
11        phoneNo: $scope.model.phoneNo
12    };
```

```

13
14     $http({
15         method: 'POST',
16         url: 'api/auth/local/join',
17         data: userInfo
18     }).then(function (response) {
19         $scope.model.emailNew = '';
20         errorModel.push({message: response.data.message, type: 'success'});
21         modalService.closeActiveWindow();
22     })['finally'](function (err) {
23         $scope.state.join = false;
24     });
25 };

```

Листинг 4: Реализация функции updatePublicationsList

```

1 $scope.updatePublicationsList = function (deep, settings) {
2     var _settings = _.defaults({
3         attr: 'year',
4         order: 'desc'
5     }, settings);
6
7     var filtered = _.filter($filter('filter')($scope.model.publications,
8         $scope.view.tableFilter),
9         publicationHelper.toFilterPredicate($scope.model.searchAttributes));
10    $scope.view.publications = _.orderBy(filtered, _settings.attr,
11        _settings.order);
12
13    if (deep) {
14        buildSearchAttributes();
15    } else {
16        updateSearchAttributes();
17    }
18    $scope.vscrollContext.container.reset();
19 };

```

Листинг 5: Реализация функции создания табличного отчета docx

```

1 $scope.compileDoc = function (data, type, authors) {
2     switch (type) {
3         case 'table':
4             var reportAuthors = _(authors).filter('isSelected').map('Name').value();
5             $scope.compiler.export.docxClusterReport(data, reportAuthors,
6                 publicationModel.shortMapping, $scope.model.categories,
7                 $scope.addToDownloadList);
8             break;
9     }

```

```
10 };
```

Листинг 6: Схема публикаций

```
1 var PublicationSchema = new Schema({
2   _owner: {type: ObjectId},
3   category: {type: String, required: true},
4   authors: [{type: String, required: true}],
5   title: {type: String, required: true},
6   year: {type: Number, required: true},
7   pagesCount: Number,
8   publicationType: String,
9   publicationLink: String,
10  comment: String,
11  firstPage: String,
12  lastPage: String,
13  journal: String,
14  journalEdition: String,
15  journalVolume: String,
16  higherCertificationCommission: {type: Boolean, default: false},
17  scopus: {type: Boolean, default: false},
18  webOfScience: {type: Boolean, default: false},
19  city: String,
20  publishingHouse: String,
21  collectionTitle: String,
22  conferenceTitle: String,
23  conferencePlace: String,
24  conferenceDate: String,
25  inventoryNumber: Number,
26  registrationNumber: Number,
27  patentNumber: Number,
28  registrationDate: String,
29  publishedDate: String,
30  reportStatus: String,
31  topic: String,
32  bulletin: String,
33  organization: String,
34  specialty: String,
35  degree: String,
36  recommendationLabel: String,
37  underEditorship: String,
38  isInEnglish: {type: Boolean, default: false}
39 });
```

Листинг 7: Пример добавления атрибута к публикации категории “Статья в журнале”

```

1 categoriesFields: {
2     journalArticle: {
3         availableAttrs: [
4             {header: 'Публикация на английском языке',
5               attr: 'isInEnglish', type: 'checkbox', required: false},
6             ...
7         ]

```

Листинг 8: Вывод количества страниц

```

1 var toPages = function (publication) {
2     var res = "";
3     if (!isEmpty(publication.firstPage)) {
4         if (publication.isInEnglish) {
5             res += "P. " + publication.firstPage;
6         }
7         else {
8             res += "C. " + publication.firstPage;
9         }
10        if (!isEmpty(publication.lastPage)) {
11            res += "-" + publication.lastPage + '.';
12        }
13    }
14    else {
15        if (!isEmpty(publication.lastPage)) {
16            if (publication.isInEnglish) {
17                res += "P. " + publication.lastPage + '.';
18            }
19            else {
20                res += "C. " + publication.lastPage + '.';
21            }
22        }
23        else {
24            if (!isEmpty(publication.pagesCount)) {
25                if (publication.isInEnglish) {
26                    res += publication.pagesCount + ' p.';
27                }
28                else {
29                    res += publication.pagesCount + ' с.';
30                }
31            }
32        }
33    }
34    return res + " ";
35 };

```

Листинг 9: Обзор функции getPublicationCategory

```

1 $scope.getPublicationCategory = function (category) {
2     var index = _.findIndex($scope.model.categories, function (item) {
3         return category &&
4             (item.Key.toLowerCase() == category.toLowerCase());
5     });
6
7     if (index > -1) {
8         return $scope.model.categories[index].Name;
9     }
10    else {
11        return false;
12    }
13 };

```

Листинг 10: Динамическое отображение пользовательской категории публикации

```

1 <div class="col s12"
2     ng-if="view.create.mappingApplied &&
3     !getPublicationCategory(view.create.publication.category)">
4     <i>
5         Не найдено соответствия Вашей категории
6         <b>{{view.create.publication.category}}</b>
7         с категориями, представленными в системе.
8         Выберите наиболее подходящую для продолжения
9     </i>
10 </div>

```

Листинг 11: Подключение библиотеки VScroll

```

1 <tbody hscroll-body vscroll vscroll-port-y="vscrollContext">
2     <tr ng-repeat="user in view.users | filter: view.filter |
3     vscroll: vscrollContext track by $index"
4     vscroll-row="{{:::$index}}">
5     <td class="chevron">
6         <input id="{{:::$index}}-users" type="checkbox" ng-model="user.isSelected"
7         groupbox-item/>
8         <label for="{{:::$index}}-users"></label>
9     </td>
10    <td ng-repeat="column in model.columns" class="{{:::column.attr}}">
11        {{user[column.attr]}}
12    </td>
13 </tr>
14 </tbody>

```

Листинг 12: Открытие модального окна сброса пароля

```

1 $scope.openResetPassword = function () {
2     $scope.modalService
3         .create($scope,
4             'app.components.admin.users.usersResetPassword.html')
5         .open();
6 };

```

Листинг 13: Функция сброса пароля пользователя

```

1 $scope.resetPassword = function () {
2     $scope.modalService.closeActiveWindow();
3
4     var query = [];
5     $scope.state.update = true;
6     _.forEach($scope.model.selectedUsers, function (item) {
7         query.push(
8             $http({
9                 method: 'POST',
10                url: '/api/auth/local/reset',
11                data: {email: item.local.email}
12            }));
13    });
14
15    $q.all(query)
16        .then(function () {
17            $scope.state.update = false;
18        });
19 };

```

Листинг 14: Функция создания группы

```

1 $scope.submit = function () {
2     $scope.state.groupAdd = true;
3
4     var data = {
5         title: $scope.model.newGroup.title,
6         description: $scope.model.newGroup.description,
7         link: $scope.model.newGroup.link
8     };
9
10    $http({
11        method: 'POST',
12        headers: {'Content-Type': 'application/json'},
13        url: 'api/groups',
14        data: data
15    }).then(function (response) {
16        errorModel.push({message: response.data.message, type: 'success'});

```

```
17     $scope.model.groups.push($scope.model.newGroup);
18     $scope.model.newGroup = {};
19     $scope.modalService.closeActiveWindow();
20   })['finally'](function (err) {
21     $scope.state.groupAdd = false;
22   });
23   };
```


Список литературы

- [1] AngularJS — Superheroic JavaScript MVW Framework. — 2017. — URL: <https://angular.io/> (online; accessed: 22.05.2017).
- [2] Bitbucket — The Git solution for professional teams. — 2017. — URL: <https://bitbucket.org/> (online; accessed: 22.05.2017).
- [3] CSS docs. — 2017. — URL: <https://developer.mozilla.org/ru/docs/Web/CSS> (online; accessed: 22.05.2017).
- [4] Express.js docs. — 2017. — URL: <http://expressjs.com/en/api.html> (online; accessed: 22.05.2017).
- [5] Git — distributed-is-the-new-centralized. — 2017. — URL: <https://git-scm.com/> (online; accessed: 22.05.2017).
- [6] GitHub — The world's leading software development platform. — 2017. — URL: <https://github.com/> (online; accessed: 22.05.2017).
- [7] Google Scholar — Bibliographic database. — 2017. — URL: <https://scholar.google.ru/> (online; accessed: 22.05.2017).
- [8] HTML docs. — 2017. — URL: <https://developer.mozilla.org/en/docs/Web/HTML> (online; accessed: 22.05.2017).
- [9] JavaScript Object Notation. — 2017. — URL: <http://json.org/> (online; accessed: 22.05.2017).
- [10] JavaScript docs. — 2017. — URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (online; accessed: 22.05.2017).
- [11] Lodash — A JavaScript utility library delivering modularity, performance, extras. — 2017. — URL: <https://lodash.com/> (online; accessed: 22.05.2017).
- [12] Material design guidelines. — 2017. — URL: <https://material.io/guidelines/> (online; accessed: 22.05.2017).

- [13] Materialize Documentation. — 2017. — URL: <http://materializecss.com/> (online; accessed: 22.05.2017).
- [14] Mendeley — Reference manager and an academic social network. — 2017. — URL: <https://www.mendeley.com/> (online; accessed: 22.05.2017).
- [15] MongoDB documentation. — 2017. — URL: <https://docs.mongodb.org/manual/> (online; accessed: 22.05.2017).
- [16] Node.js. — 2017. — URL: <https://nodejs.org/en/> (online; accessed: 22.05.2017).
- [17] Papa Parse — Powerful CSV Parser for JavaScript. — 2017. — URL: <http://papaparse.com/> (online; accessed: 22.05.2017).
- [18] ResearchGate is a social networking site for scientists and researchers. — 2017. — URL: <https://www.researchgate.net/> (online; accessed: 22.05.2017).
- [19] SASS docs. — 2017. — URL: http://sass-lang.com/documentation/file.SASS_REFERENCE.html (online; accessed: 22.05.2017).
- [20] Scopus is the largest abstract and citation database of peer-reviewed literature. — 2017. — URL: <https://www.scopus.com/> (online; accessed: 22.05.2017).
- [21] SourceTree — Free Git and Hg Client for Mac and Windows. — 2017. — URL: <https://www.sourcetreeapp.com/> (online; accessed: 22.05.2017).
- [22] VScroll 1.3 + angularjs. — 2017. — URL: <https://github.com/klumba12/vscroll> (online; accessed: 22.05.2017).
- [23] W3C: Document Object Model. — 2017. — URL: <https://www.w3.org/TR/dom/> (online; accessed: 22.05.2017).

- [24] Web of Science. — 2017. — URL: <http://ipscience.thomsonreuters.com/product/web-of-science/> (online; accessed: 22.05.2017).
- [25] WebStorm: The Smartest JavaScript IDE. — 2017. — URL: <https://www.jetbrains.com/webstorm/> (online; accessed: 22.05.2017).
- [26] bcrypt docs. — 2017. — URL: <https://www.npmjs.com/package/bcrypt> (online; accessed: 22.05.2017).
- [27] eLIBRARY.RU — Научная электронная библиотека. — 2017. — URL: <http://elibrary.ru/> (дата обращения: 22.05.2017).
- [28] Проект RADOMS. — 2017. — URL: <http://radoms.ru> (дата обращения: 22.05.2017).
- [29] РИНЦ. — 2017. — URL: http://elibrary.ru/projects/citation/cit_index.asp (дата обращения: 22.05.2017).